A License-Based Search Engine

Benjamin Moreau, Patricia Serrano-Alvarado, Matthieu Perrin, Emmanuel Desmontils

A License-Based Search Engine

Benjamin Moreau[1,2], Patricia Serrano-Alvarado[1], Matthieu Perrin[1], and Emmanuel Desmontils[1]

[1] Nantes University, LS2N, CNRS, UMR6004, 44000 Nantes, France
{Name.LastName@}univ-nantes.fr
[2] OpenDataSoft {Name.Lastname}@opendatasoft.com

Abstract. The reuse of licensed resources to produce new ones is very common and encouraged on the Web. But producing resources whose licenses are compliant with all reused resource licenses is not easy. It is necessary to know (1) the set of licenses with which the license of the produced resource is compliant and (2) what are the available resources whose licenses are part of this set. With CaLi, we provide an answer to the first concern. CaLi is a lattice-based model that partially orders licenses in terms of compatibility and compliance. In this demonstration, we illustrate the usability of CaLi through a prototype for the second concern. That is, based on a CaLi ordering of licenses we implement a license-based search engine which can answer questions such as *"find licensed resources that can be reused under a given license"* or *"find licensed resources that can reuse a resource that has a particular license"*.

1 Introduction and motivation

To facilitate reuse on the Web, resource producers should systematically associate licenses with resources before sharing or publishing them [3]. Licenses specify precisely the conditions of reuse of resources, i.e., what actions are *permitted*, *obliged* and *prohibited* when using the resource.

For a resource producer, choosing the appropriate license for a combined resource or choosing the appropriate licensed resources for a combination involves choosing a license compliant with all the licenses of combined resources as well as analysing the reusability of the resulting resource through the compatibility of its license.

We consider simplified definitions of compliance and compatibility [1], *a license l_j is compliant with a license l_i if a resource licensed under l_i can be licensed under l_j without violating l_i*. If a license l_j is compliant with l_i then we consider that l_i is compatible with l_j and that resources licensed under l_i are reusable with resources licensed under l_j. In general, if l_i is compatible with l_j then l_j is more (or equally) restrictive than l_i. We also consider that *a license l_j is more (or equally) restrictive than a license l_i if l_j allows at most the same permissions and has at least the same prohibitions/obligations than l_i*.

But producing resources whose licenses are compliant with all reused resource licenses is difficult. It is necessary to know (1) the set of licenses with which the

license of the produced resource is compliant and (2) what are the pertinent and available resources whose licenses are part of this set.

With CaLi [1], we provide an answer to the first concern. CaLi is a lattice-based model to define compatibility and compliance relations among licenses. It is based on a restrictiveness relation that is refined with constraints to take into account the semantics of actions existing in licenses.

For the second concern, imagine a license-based search engine that can answer questions such as "*find all resources that can be reused under the CC BY-NC license*". The answer must contain resources licensed under licenses such as CC BY and CC BY-NC itself that are less or as restrictive as CC BY-NC and compatible with it.

There exist search engines in services such as GitHub[3], APISearch[4], CC search[5], LODAtlas[6], DataHub[7], Google Dataset Search[8] or OpenDataSoft[9] that can find resources licensed under a particular license. However they can not find resources whose licenses are compatible or compliant with a particular license.

We illustrate the usability of CaLi by answering the second concern. We developed a prototype of a search engine based on a CaLi ordering of licenses, $ODRL\_CaLi$. The goal is to be able find resources whose licenses are compatible or compliant with a particular license. Our prototype can answer questions such as: *"find licensed resources that can be reused under a given license"* or *"find licensed resources that can reuse a resource that has a particular license"*.

In our search engine, resources (linked data and source code) are associated to licenses. Licenses are described in RDF with the ODRL vocabulary[10] and ordered in terms of compatibility according to the $ODRL\_CaLi$ ordering. In addition to indexing licenses, the titles, descriptions and uri of each licensed resources are also indexed to enable full-text search. We remark that we are not interested in implementing ODRL. We use the ODRL vocabulary because it is the most complete vocabulary for licenses and it is well accepted by the community.

In the following, Section 2 overviews the CaLi model and the $ODRL\_CaLi$ ordering used in our search engine, and Section 3 describes the demonstration.

2 Modelling the compatibility of licenses

Inspired by lattice-based access control models, we propose a CaLi model as a tuple $\langle \mathcal{A}, \mathcal{LS}, C_{\mathcal{L}}, C_{\rightarrow} \rangle$ that partially orders licenses, such that [1]:

1. $\mathcal{A}$ is a set of *actions* (e.g., *read, modify, distribute*, etc.);

[3] `https://github.com/`
[4] `http://apis.io/`
[5] `https://ccsearch.creativecommons.org/`
[6] `http://lodatlas.lri.fr/`
[7] `https://datahub.io/`
[8] `https://toolbox.google.com/datasetsearch`
[9] `https://data.opendatasoft.com/`
[10] `https://www.w3.org/TR/odrl-model/`

2. $\mathcal{LS}$ is a *restrictiveness lattice of status* that defines (i) all possible status (e.g., permissions, obligations, prohibitions, etc.) of an action in a license and (ii) the restrictiveness relation among status denoted by $\leqslant_S$;
3. $C_{\rightarrow}$ is a set of *compatibility constraints* to identify if a restrictiveness relation between two licenses is also a compatibility relation; and
4. $C_{\mathcal{L}}$ is a set of *license constraints* to identify non-valid licenses.

In CaLi, $\mathcal{L}_{\mathcal{A},\mathcal{LS}}$ defines the set of all licenses that can be expressed with $\mathcal{A}$ and $\mathcal{LS}$. $(\mathcal{L}_{\mathcal{A},\mathcal{LS}}, \leqslant_{\mathcal{R}})$ is the restrictiveness lattice of licenses that defines the restrictiveness relation $\leqslant_{\mathcal{R}}$ over the set of all licenses $\mathcal{L}_{\mathcal{A},\mathcal{LS}}$. With $C_{\mathcal{L}}$ non-valid licenses are identified. We consider a license l_i as non-valid if a resource can not be licensed under l_i. If two valid licenses have a restrictiveness relation then it is possible that they have a compatibility relation too. To identify the compatibility among licenses, CaLi refines the restrictiveness relation with compatibility constraints $C_{\rightarrow}$.

$ODRL\_CaLi$, is a CaLi ordering $\langle \mathcal{A}, \mathcal{LS}, C_{\mathcal{L}}, C_{\rightarrow} \rangle$ such that:

- $\mathcal{A}$ is the set of 72 actions considered by ODRL[11];
- $\mathcal{LS}$ is the restrictiveness lattice of status where (i) the possible status are Permission, Duty, Prohibition[12] or Undefined (for actions that do not appear in the license), and (ii) the restrictiveness relation is $Undefined \leqslant_S Permission \leqslant_S Duty \leqslant_S Prohibition$; and
- $C_{\mathcal{L}}, C_{\rightarrow}$ are the sets of constraints, inspired from the ODRL information model, defined below.

$C_{\mathcal{L}} = \{\omega_{\mathcal{L}_1}, \omega_{\mathcal{L}_2}\, \omega_{\mathcal{L}_3}\}$ allows to invalidate a license (1) when *cc:CommercialUse* is required, (2) when *cc:ShareAlike* is prohibited and (3) when the semantics of a permitted or obliged action is included in a prohibited action (e.g. if *CommercialUse* is permitted then *use* should not be prohibited because *CommercialUse* implies *use*):

$$\omega_{\mathcal{L}_1}(l_i) = \begin{cases} False \text{ if } l_i(\textit{cc:CommercialUse}) = \text{Duty}; \\ True \text{ otherwise.} \end{cases}$$

$$\omega_{\mathcal{L}_2}(l_i) = \begin{cases} False \text{ if } l_i(\textit{cc:ShareAlike}) = \text{Prohibition}; \\ True \text{ otherwise.} \end{cases}$$

$$\omega_{\mathcal{L}_3}(l_i) = \begin{cases} False \text{ if } a_i \text{ odrl:includedIn } a_j \\ \qquad \text{AND } (l_i(a_i) = \text{Permitted OR } l_i(a_i) = \text{Duty}) \\ \qquad \text{AND } l_i(a_j) = \text{Prohibited}; \\ True \text{ otherwise.} \end{cases}$$

$C_{\rightarrow} = \{\omega_{\rightarrow_1}, \omega_{\rightarrow_2}\}$ allows to identify (1) when *cc:ShareAlike* is required and (2) when *cc:DerivativeWorks* is prohibited. That is because *cc:ShareAlike* requires that the distribution of derivative works be under the same license only, and *cc:DerivativeWorks*, when prohibited, does not allow the distribution of a derivative resource, regardless of the license.

$$\omega_{\rightarrow_1}(l_i, l_j) = \begin{cases} False \text{ if } l_i(\textit{cc:ShareAlike}) = \text{Duty}; \\ True \text{ otherwise.} \end{cases}$$

[11] `https://www.w3.org/TR/odrl-vocab/#actionConcepts`
[12] `https://www.w3.org/TR/odrl-model/#rule`

$$\omega_{\rightarrow_2}(l_i, l_j) = \begin{cases} False \text{ if } l_i(cc{:}DerivativeWorks) = \text{Prohibition;} \\ True \text{ otherwise.} \end{cases}$$

Other constraints could be defined to be closer to the ODRL information model but for the purposes of this demonstration these constraints are enough.

The size growth of CaLi orderings is exponential $|\mathcal{LS}|^{|\mathcal{A}|}$, so the size of $ODRL\_CaLi$ is 4^{72}, which makes it impossible to build. Nevertheless, it is not necessary to explicitly build a lattice to use it. Our search engine uses a sorting algorithm that can sort any set of licenses according to the $\mathcal{LS}$ defined above, in approximatively $n^2/2$ comparisons of restrictiveness, n being the number of licenses to sort, i.e., $O(n^2)$. This algorithm is able to insert a license in a graph in linear time $O(n)$ without sorting again the graph (see [1] for more details). Thus, our algorithm produces compatibility graphs of licenses conform to the $ODRL\_CaLi$ ordering of licenses. This algorithm is available on GitHub under the MIT license[13].

3 Demonstration

Using $ODRL\_CaLi$ and the sorting algorithm described in the previous section, we generated two compatibility graphs of licences. One for licenses that are the most used in DataHub[14] and another for the most used licenses in GitHub. Licenses are in RDF. We use the dataset of licenses proposed by [2].

Resources associated to licenses refer to some licensed RDF datasets from DataHub, from OpenDataSoft [15] and from licensed repositories from GitHub.

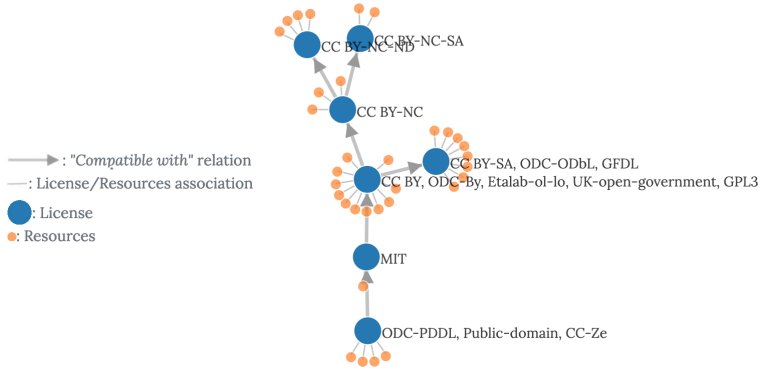
The source code of the search engine is available on GitHub[16] under the MIT license. Our demonstration is available online at `http://cali.priloo.univ-nantes.fr`.

Both compatibility graphs of licences are visually available. Figure 1a shows the compatibility graph of the CaLi ordering for some licensed RDF datasets. Blue nodes are licenses, grey arrows are compatibility relations among licenses and orange nodes are RDF datasets associated to licenses. Licenses that have the same actions in the same status are represented in the same node. In the graph, licenses that are compatible with a particular license l_i are below l_i and licenses that are compliant with l_i are above l_i. We recall that the ordering relations of compatibility and compliance that we define are reflexive, transitive and asymmetric.

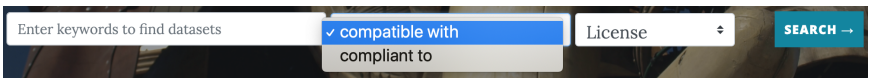
During the demonstration, attendees will be able to search for resources licensed under licenses compliant or compatible with a particular license. Figure 1b shows the search bar of our search engine. It enables full-text and license-compliant searches over each graph, for RDF datasets[17] or repositories[18]. For

[13] `https://github.com/benjimor/CaLi`
[14] `https://old.datahub.io/`
[15] `https://data.opendatasoft.com/pages/home/`
[16] `https://github.com/benjimor/CaLi-Search-Engine`
[17] `http://cali.priloo.univ-nantes.fr/ld/`
[18] `http://cali.priloo.univ-nantes.fr/rep/`

example, users can search for *datasets about 'bikes' whose licenses are compatible with the CC BY-NC license* (i.e. datasets about 'bikes' that can be reused under the CC BY-NC license). The result contains all RDF datasets indexed in the search engine where title or description contains the word 'bikes' and whose license is compatible with CC BY-NC (e.g. CC BY, MIT, CC-Ze, etc.).



(a) Compatibility graph of the $ODRL\_CaLi$ ordering for some licensed RDF datasets.



(b) Search bar of the license-based search engine.

Fig. 1: Screenshoots of the license-based search engine.

Both compatibility graphs of licences are available online through a documented API. Finally, these graphs are also accessible through a TPF server[19][20] or can be exported in RDF (turtle, xml, n3 and json-ld).

A possible extension of our search engine is to allow the collaborative addition of licenses and licensed resources. That is, to allow users to add new licenses and resources to increase the size and therefore the interest of these two graphs.

References

1. Benjamin, M., Serrano-Alvarado, P., Perrin, M., Desmontils, E.: Modelling the Compatibility of Licenses. In: Extended Semantic Web Conference (ESWC) (2019)
2. Rodríguez Doncel, V., Gómez-Pérez, A., Villata, S.: A Dataset of RDF Licenses. In: Legal Knowledge and Information Systems Conference (ICLKIS) (2014)
3. Seneviratne, O., Kagal, L., Berners-Lee, T.: Policy-Aware Content Reuse on the Web. In: International Semantic Web Conference (ISWC) (2009)

[19] `http://cali.priloo.univ-nantes.fr/api/ld/tpf`
[20] `http://cali.priloo.univ-nantes.fr/api/rep/tpf`